Fig. 1

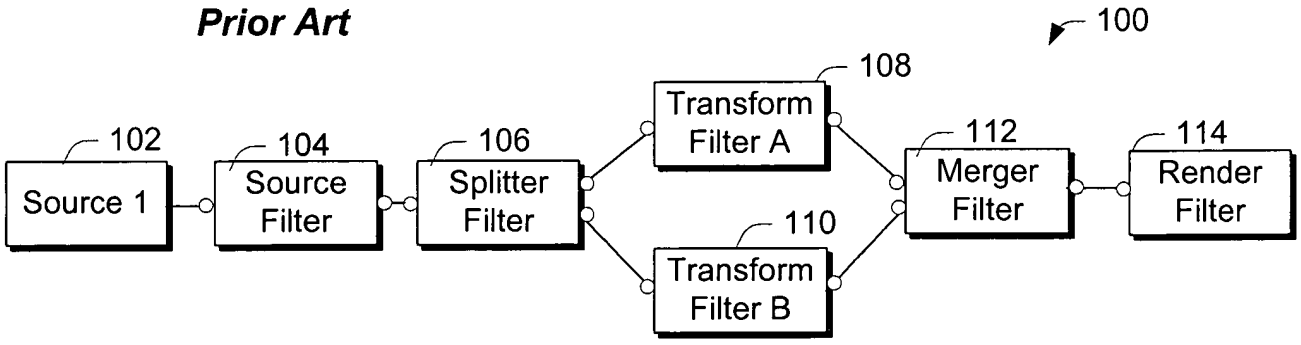
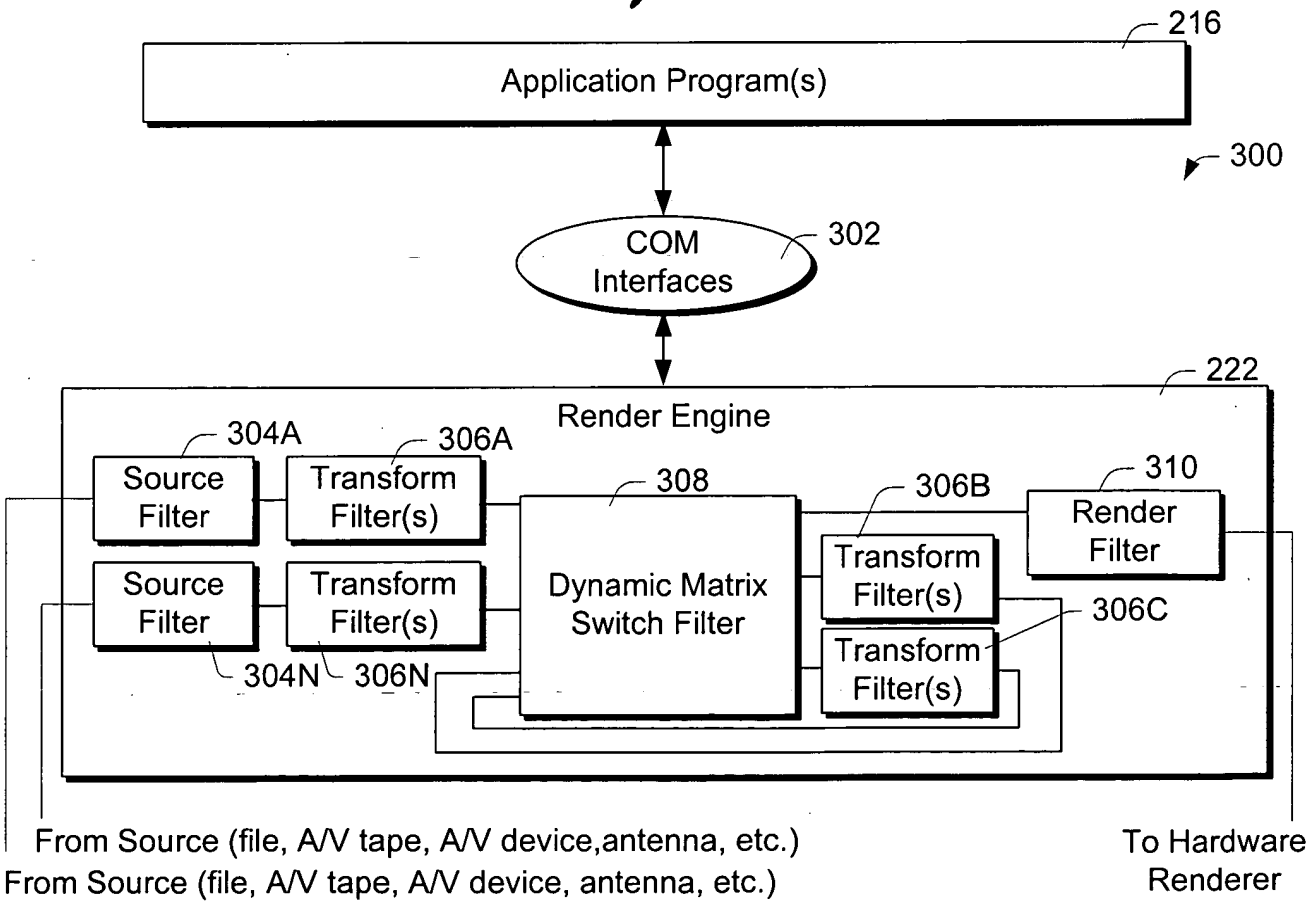

Prior Art
100
102
Source 1
104
Source Filter
106
Splitter Filter
108
Transform Filter A
110
Transform Filter B
112
Merger Filter
114
Render Filter

Fig. 3


216
Application Program(s)
300
COM Interfaces
302
222
Render Engine
304A
Source Filter
306A
Transform Filter(s)
Source Filter
Transform Filter(s)
304N
306N
308
Dynamic Matrix Switch Filter
306B
Transform Filter(s)
Transform Filter(s)
306C
310
Render Filter
From Source (file, A/V tape, A/V device,antenna, etc.)
From Source (file, A/V tape, A/V device, antenna, etc.)
To Hardware Renderer

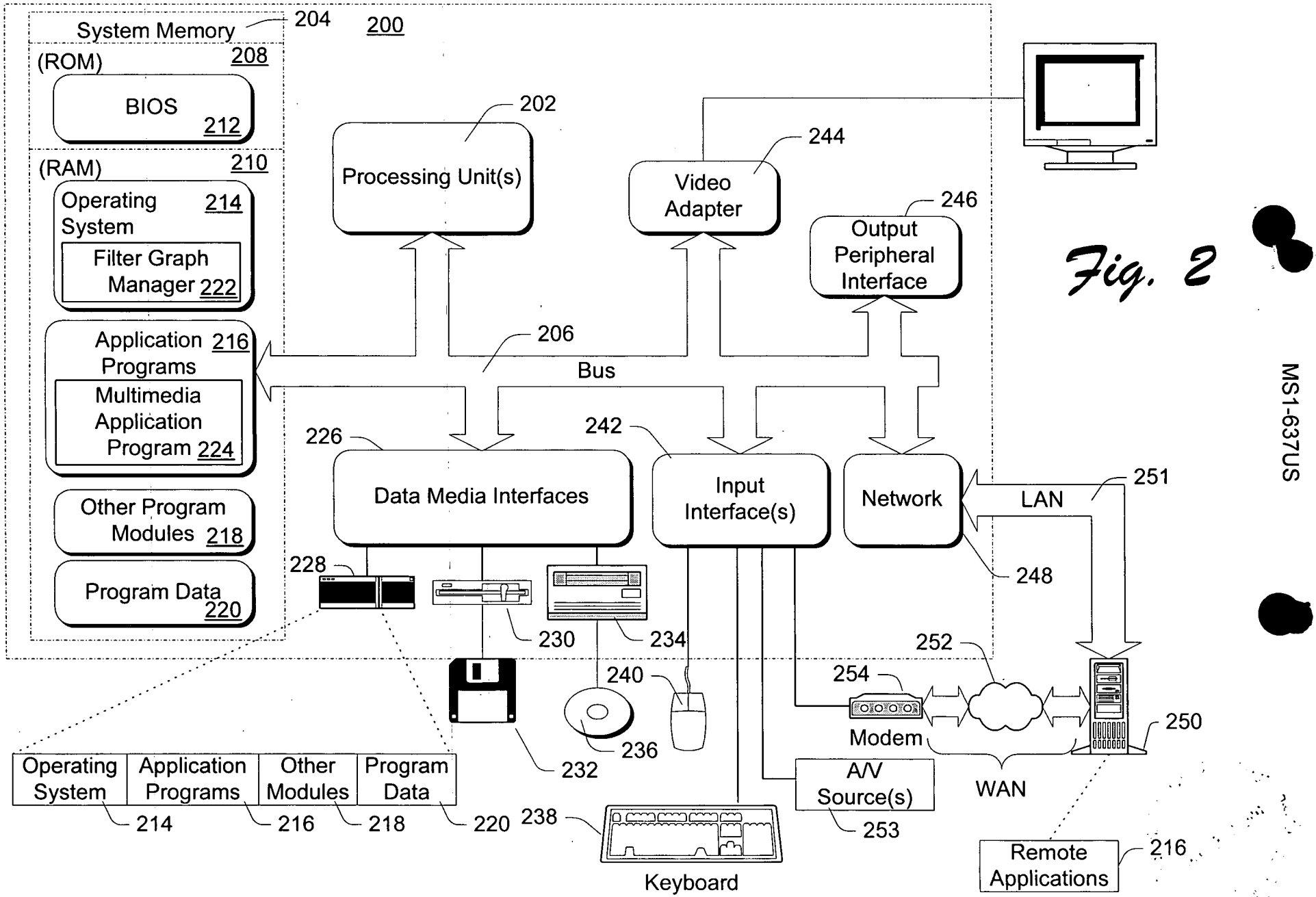
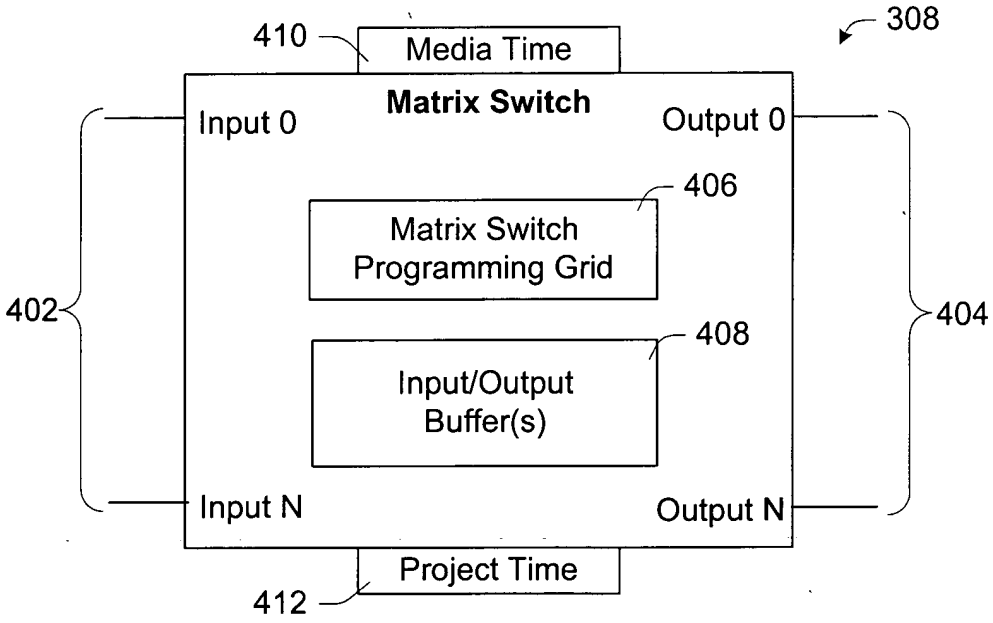


Fig. 2

MS1-637US

Fig. 4


308
410
Media Time
Matrix Switch
Input 0
Output 0
406
Matrix Switch
Programming Grid
402
404
408
Input/Output
Buffer(s)
Input N
Output N
Project Time
412

TOSED TEST269

Fig. 5

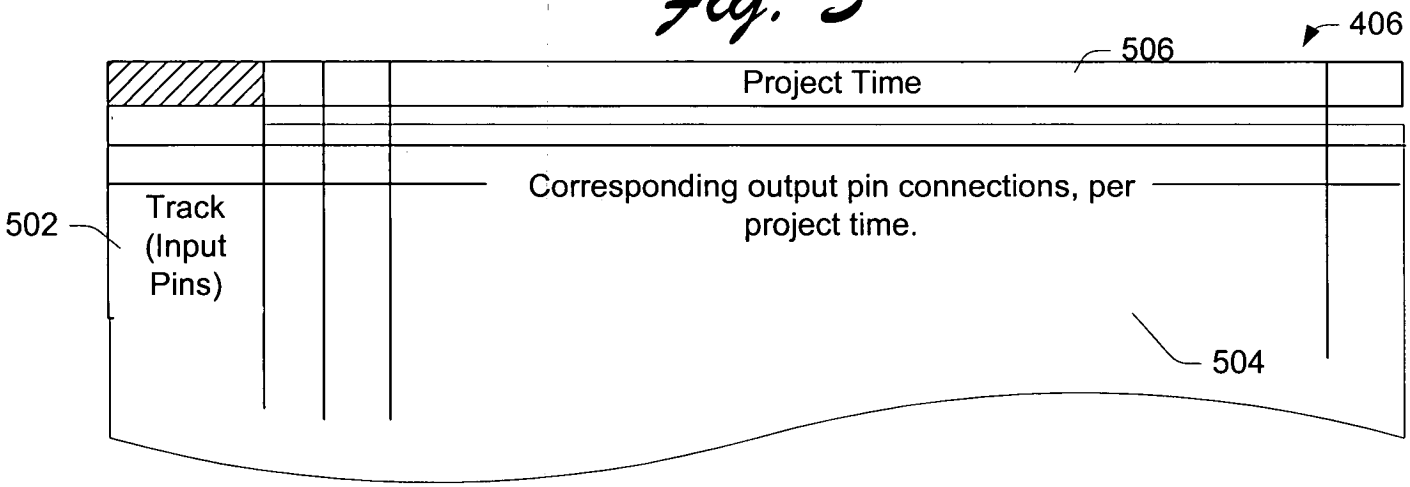


Fig. 6

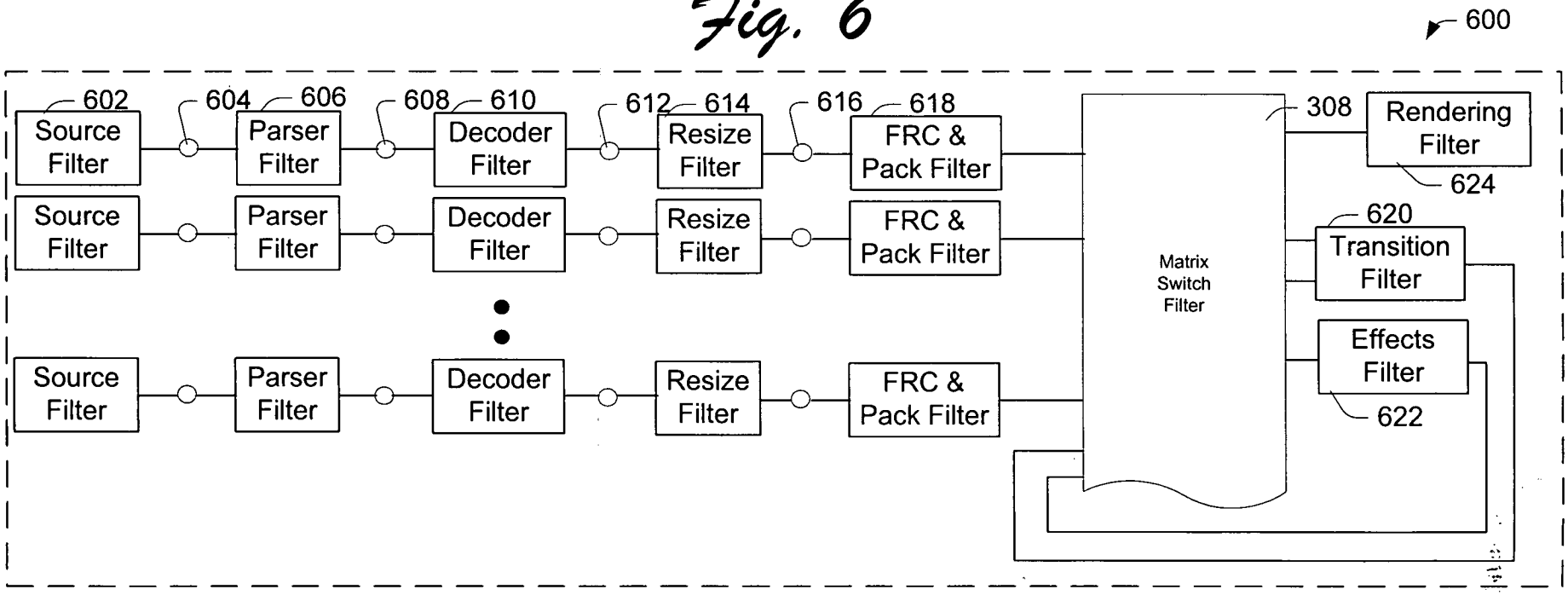


Fig. 7

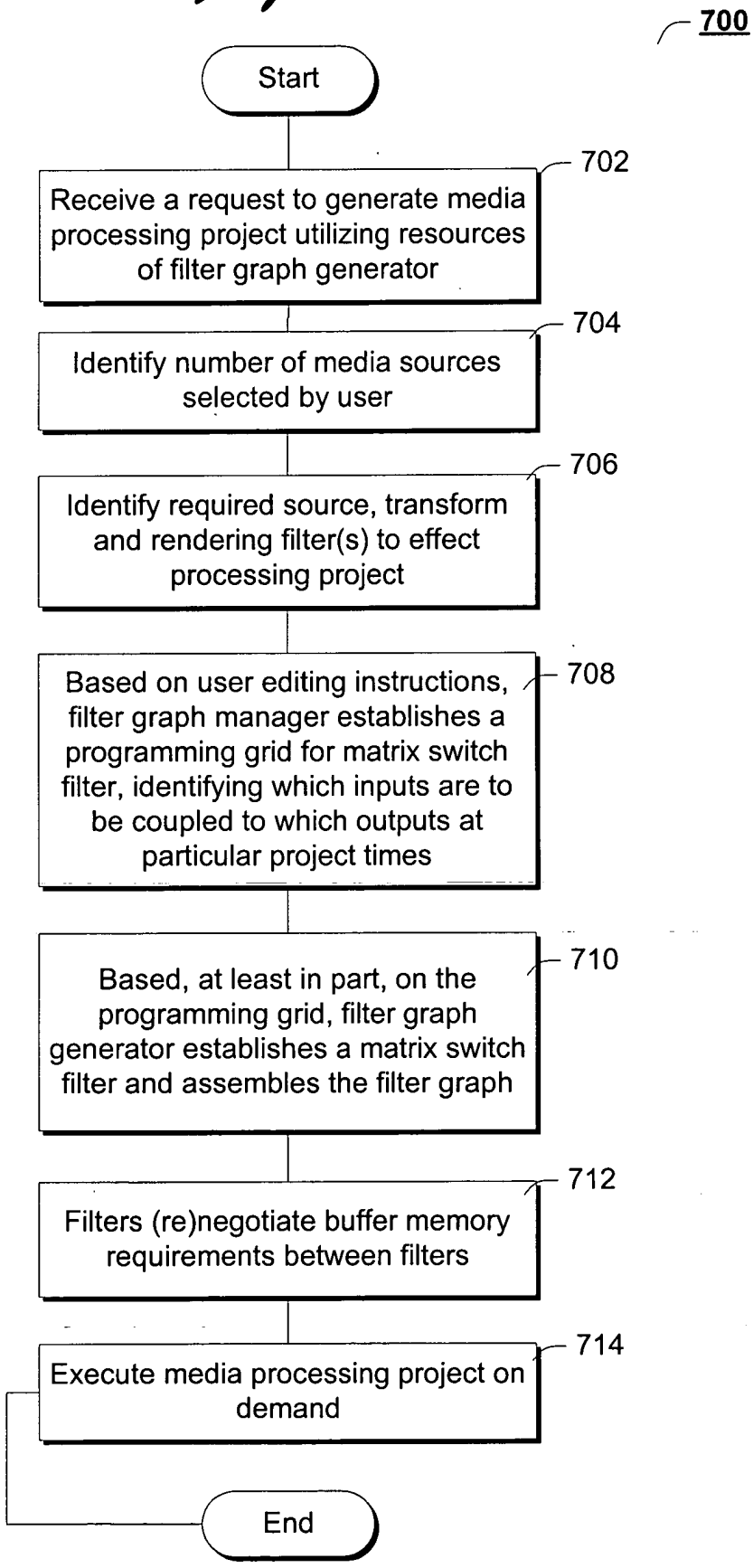
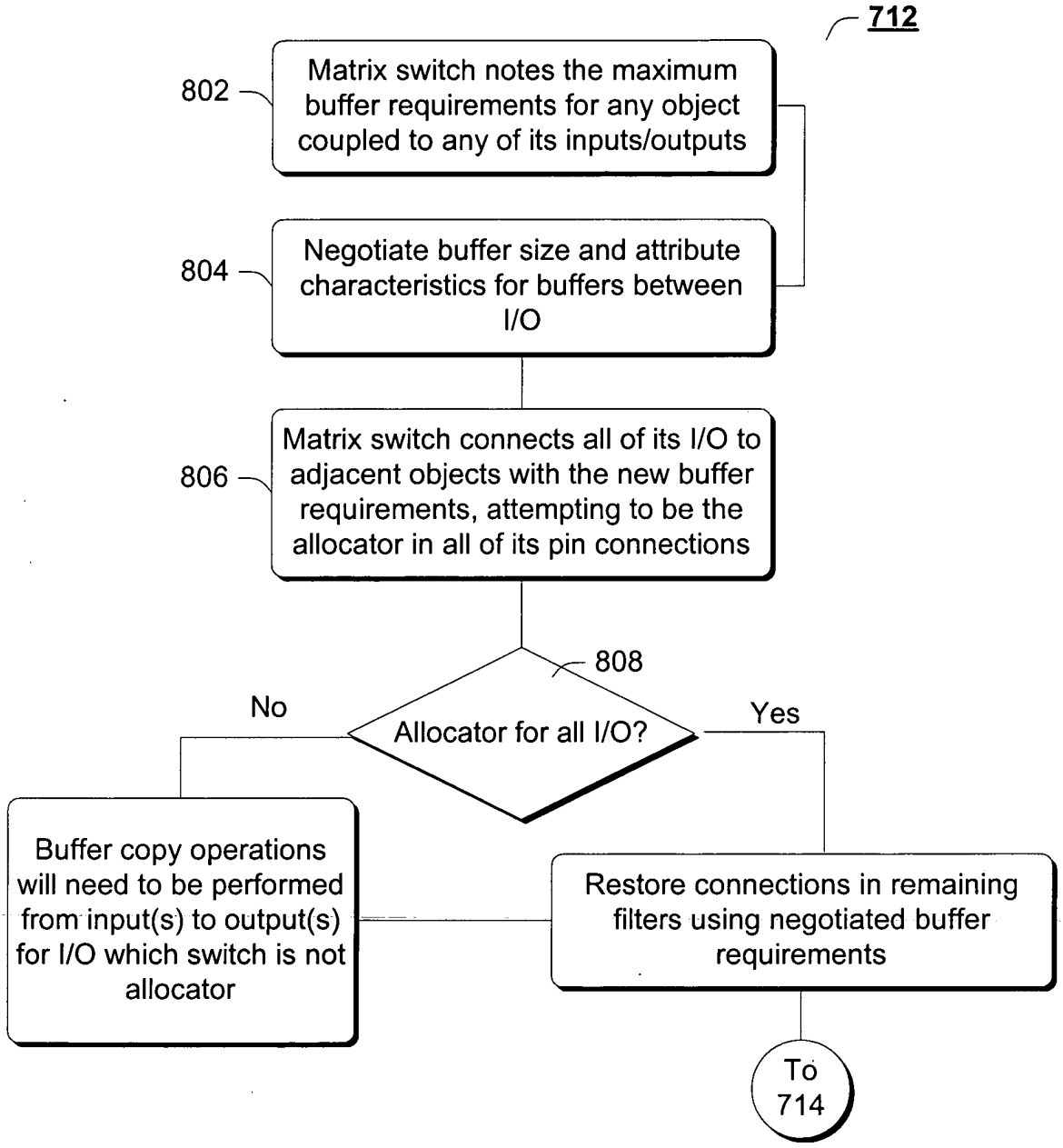

700
Start
Receive a request to generate media processing project utilizing resources of filter graph generator
702
Identify number of media sources selected by user
704
Identify required source, transform and rendering filter(s) to effect processing project
706
Based on user editing instructions, filter graph manager establishes a programming grid for matrix switch filter, identifying which inputs are to be coupled to which outputs at particular project times
708
Based, at least in part, on the programming grid, filter graph generator establishes a matrix switch filter and assembles the filter graph
710
Filters (re)negotiate buffer memory requirements between filters
712
Execute media processing project on demand
714
End

Fig. 8


712
802
Matrix switch notes the maximum buffer requirements for any object coupled to any of its inputs/outputs
804
Negotiate buffer size and attribute characteristics for buffers between I/O
806
Matrix switch connects all of its I/O to adjacent objects with the new buffer requirements, attempting to be the allocator in all of its pin connections
808
Allocator for all I/O?
No
Yes
Buffer copy operations will need to be performed from input(s) to output(s) for I/O which switch is not allocator
Restore connections in remaining filters using negotiated buffer requirements
To 714

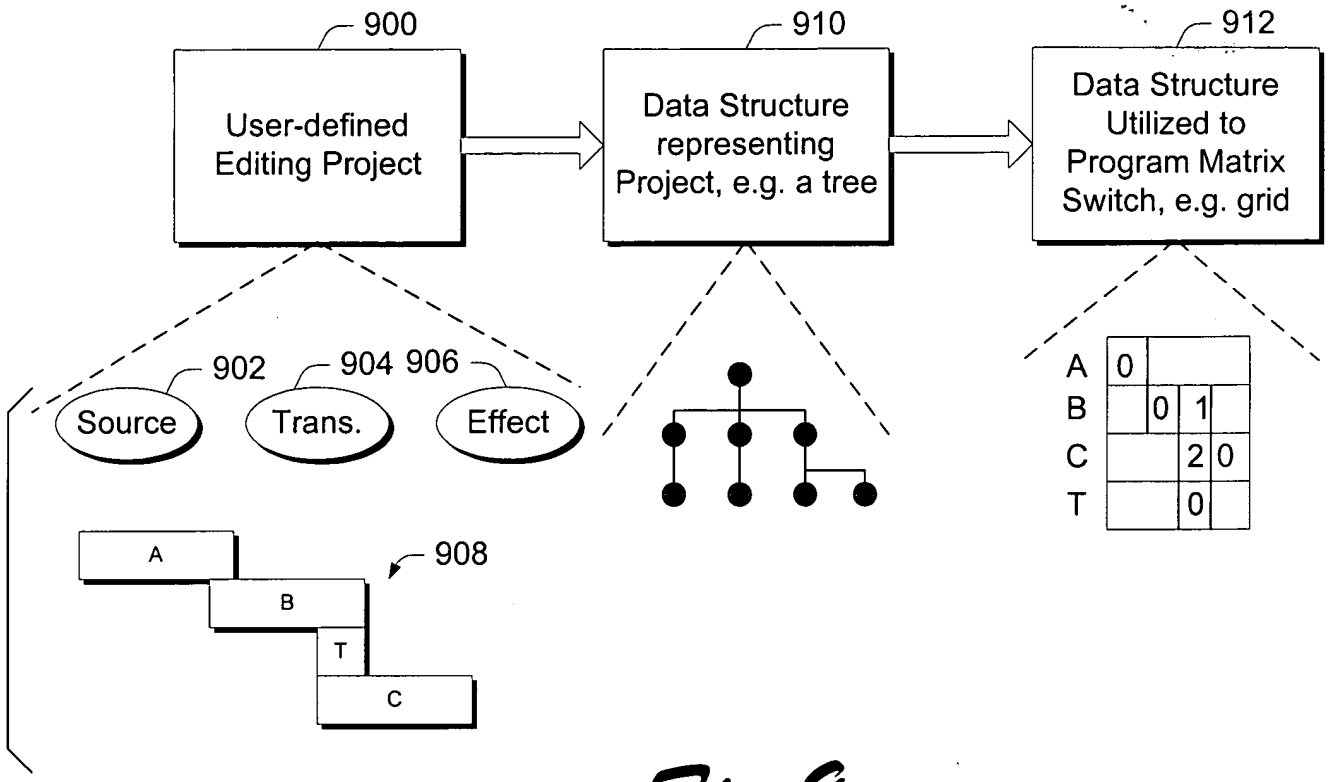

900
User-defined Editing Project
910
Data Structure representing Project, e.g. a tree
912
Data Structure Utilized to Program Matrix Switch, e.g. grid
902
904
906
Source
Trans.
Effect
A
B
T
C
908
A 0
B 0 1
C 2 0
T 0

Fig. 9

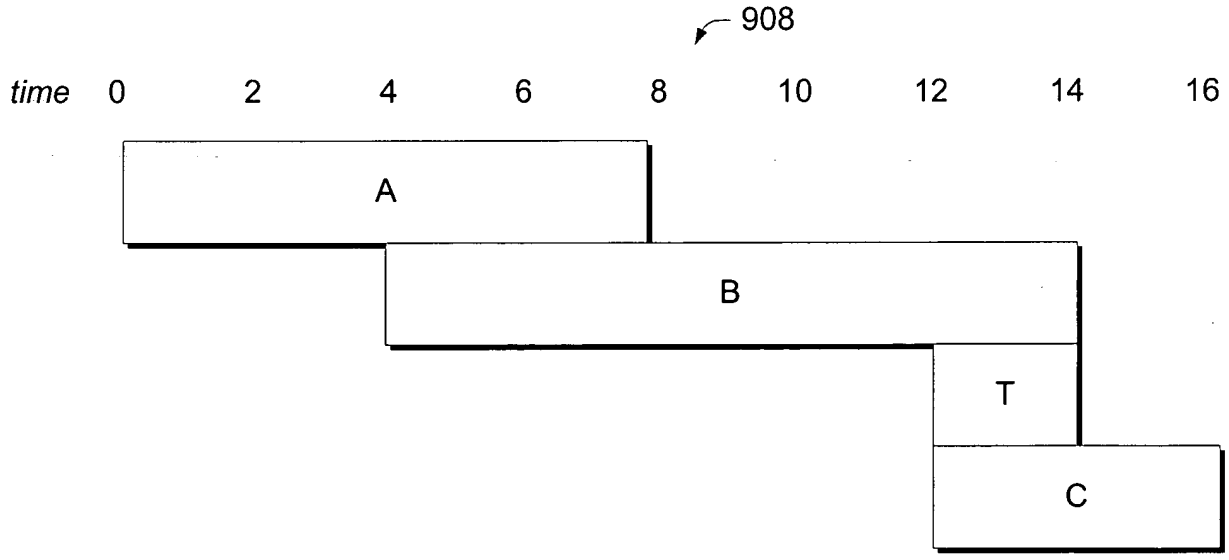

908
time 0 2 4 6 8 10 12 14 16
A
B
T
C

Fig. 10

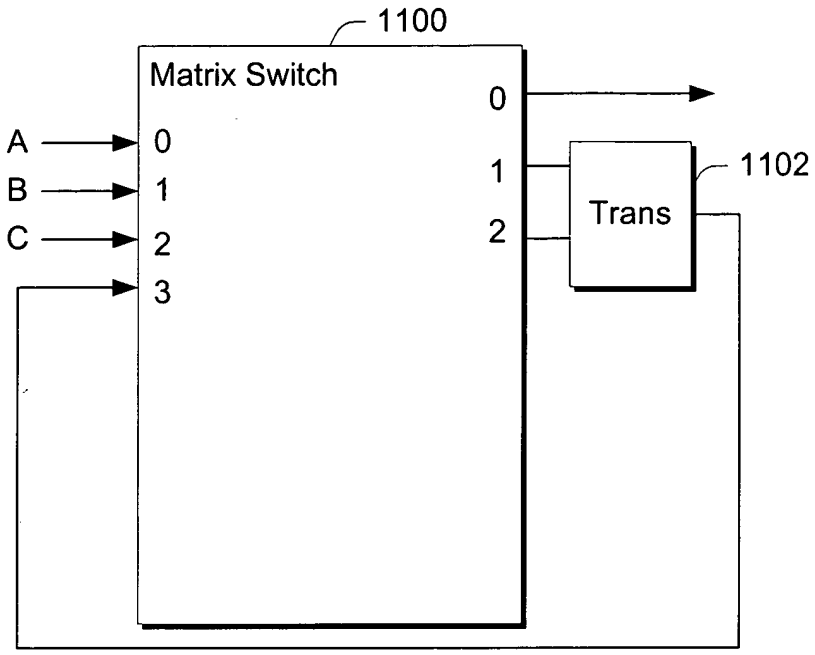

1100
Matrix Switch
A
B
C
0
1
2
3
0
1
2
Trans
1102

Fig. 11

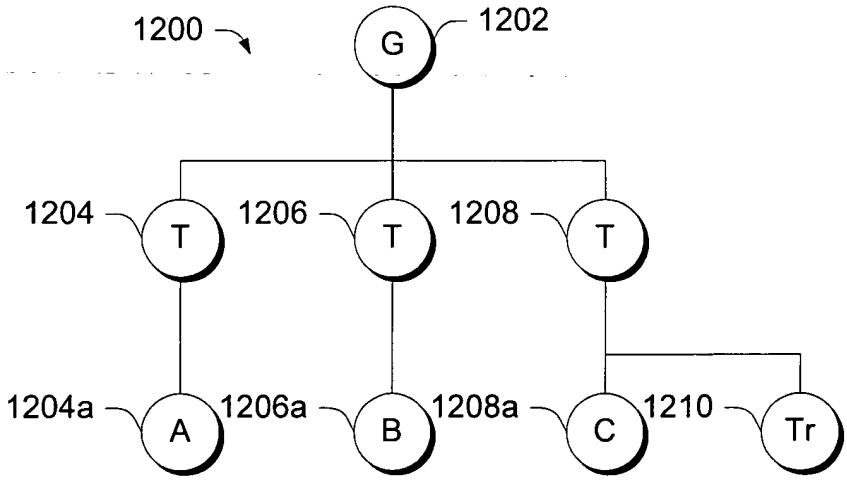

1200
G
1202
1204
T
1206
T
1208
T
1204a
A
1206a
B
1208a
C
1210
Tr

Fig. 12

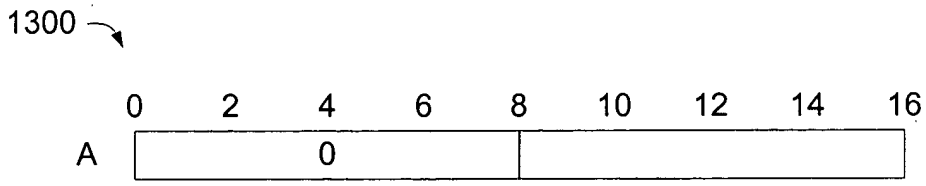


Fig. 13

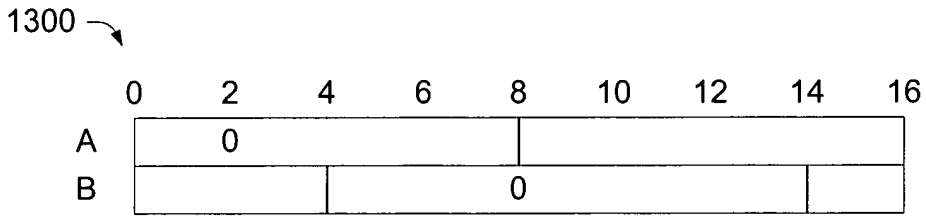


Fig. 14

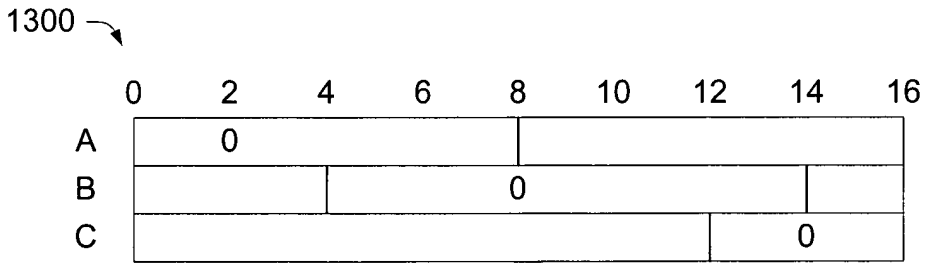


Fig. 15

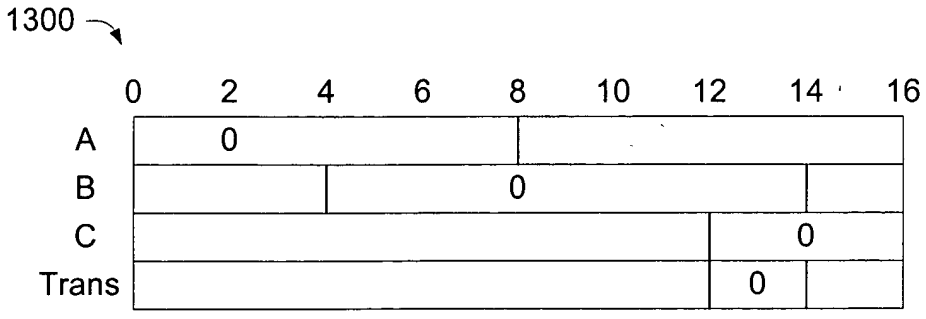


Fig. 16

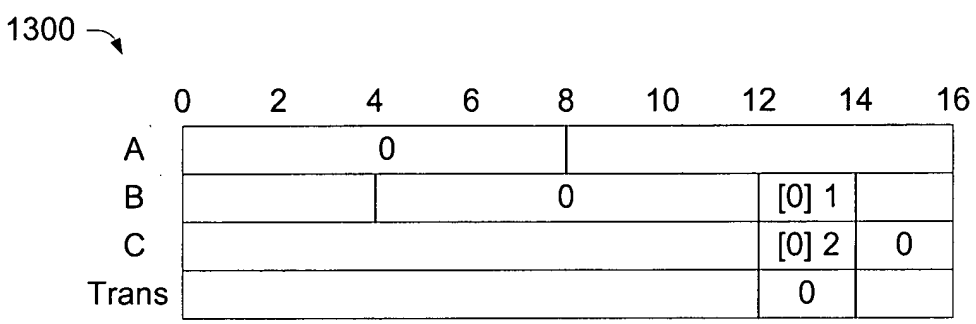

1300
0 2 4 6 8 10 12 14 16
A
B
C
Trans
0
0
[0] 1
[0] 2
0
0

Fig. 17

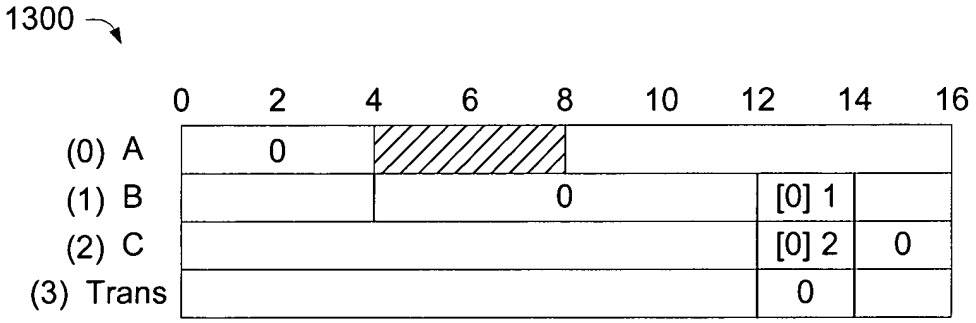

1300
0 2 4 6 8 10 12 14 16
(0) A
(1) B
(2) C
(3) Trans
0
0
[0] 1
[0] 2
0
0

Fig. 18

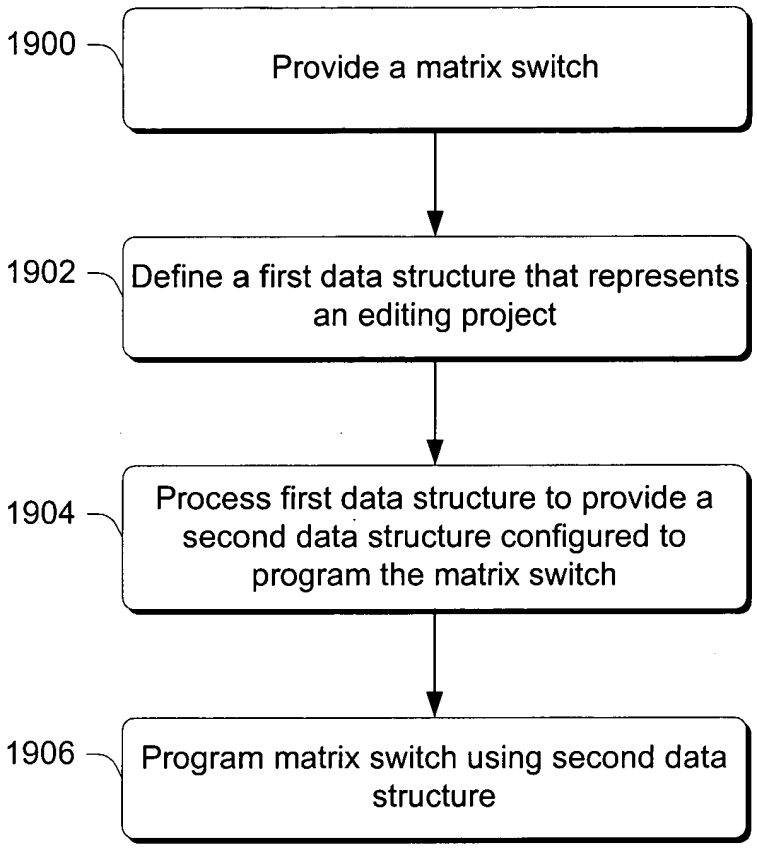

1900
Provide a matrix switch
1902
Define a first data structure that represents an editing project
1904
Process first data structure to provide a second data structure configured to program the matrix switch
1906
Program matrix switch using second data structure

Fig. 19

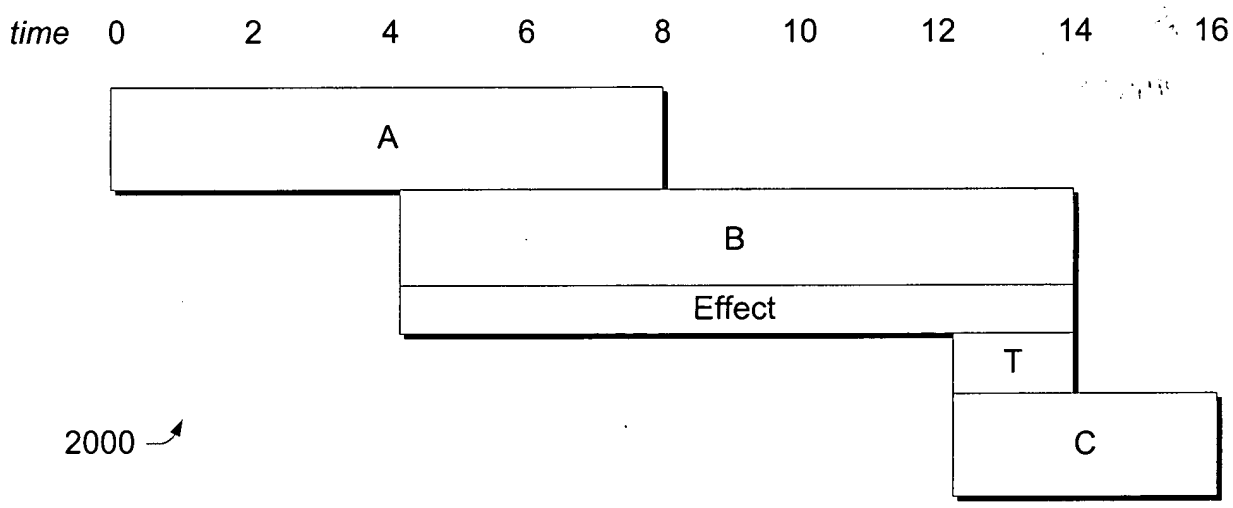


Fig. 20

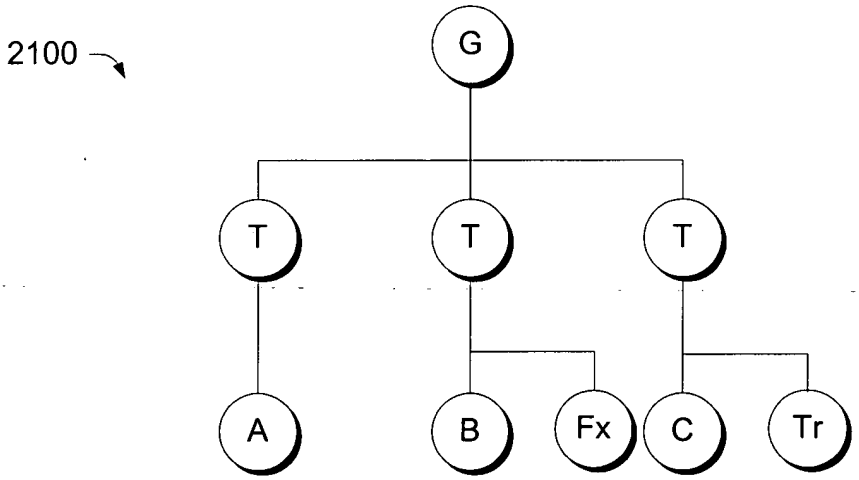


Fig. 21

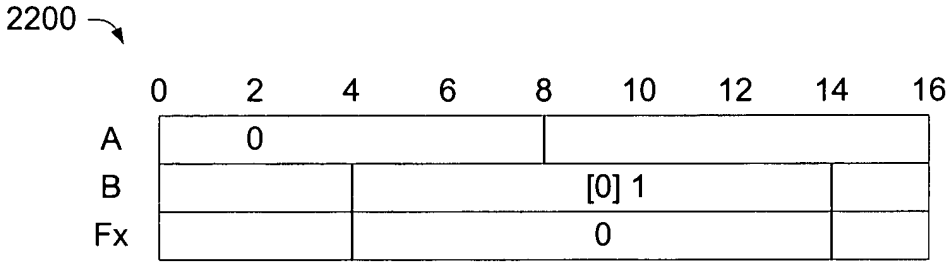


Fig. 22

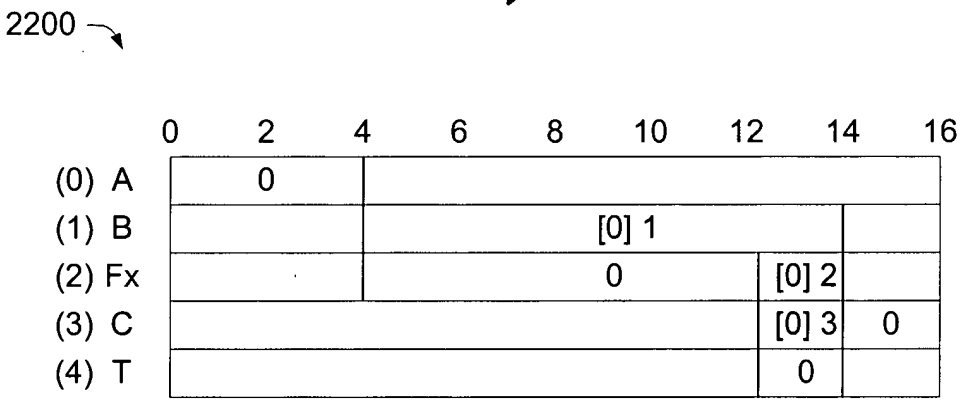


Fig. 23

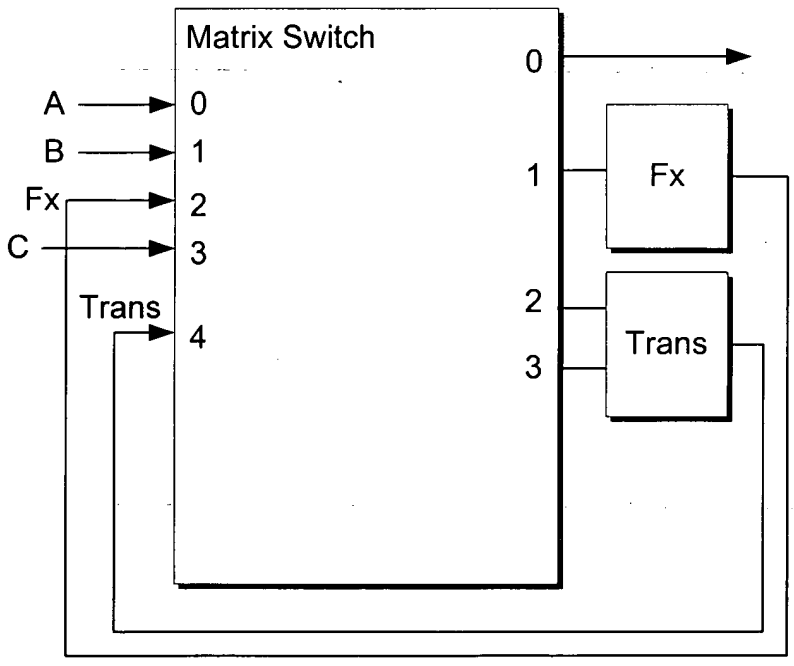


Fig. 24

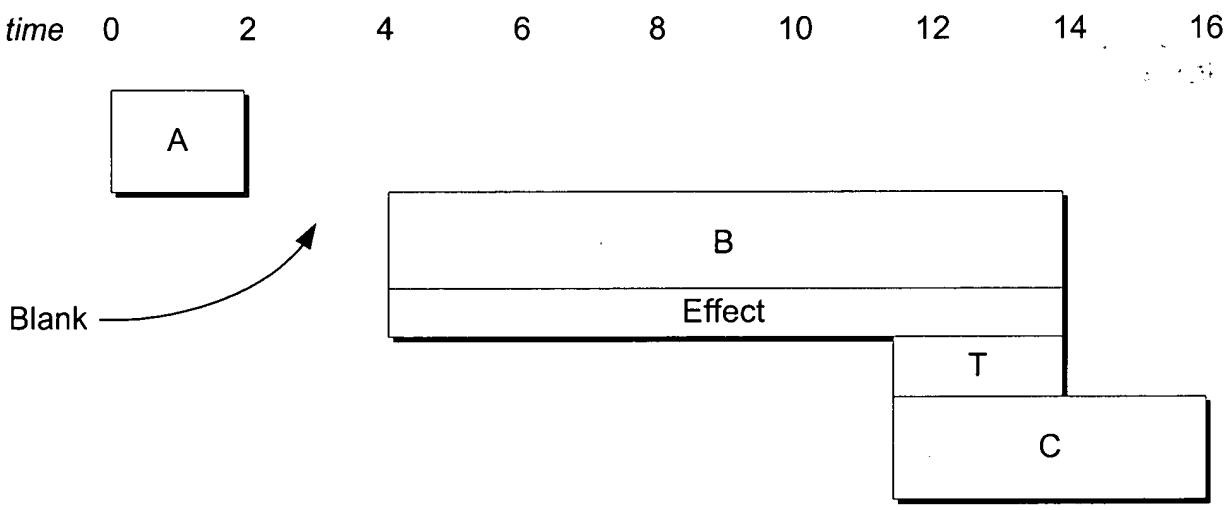

time
0
2
4
6
8
10
12
14
16
A
B
Effect
T
C
Blank

Fig. 25

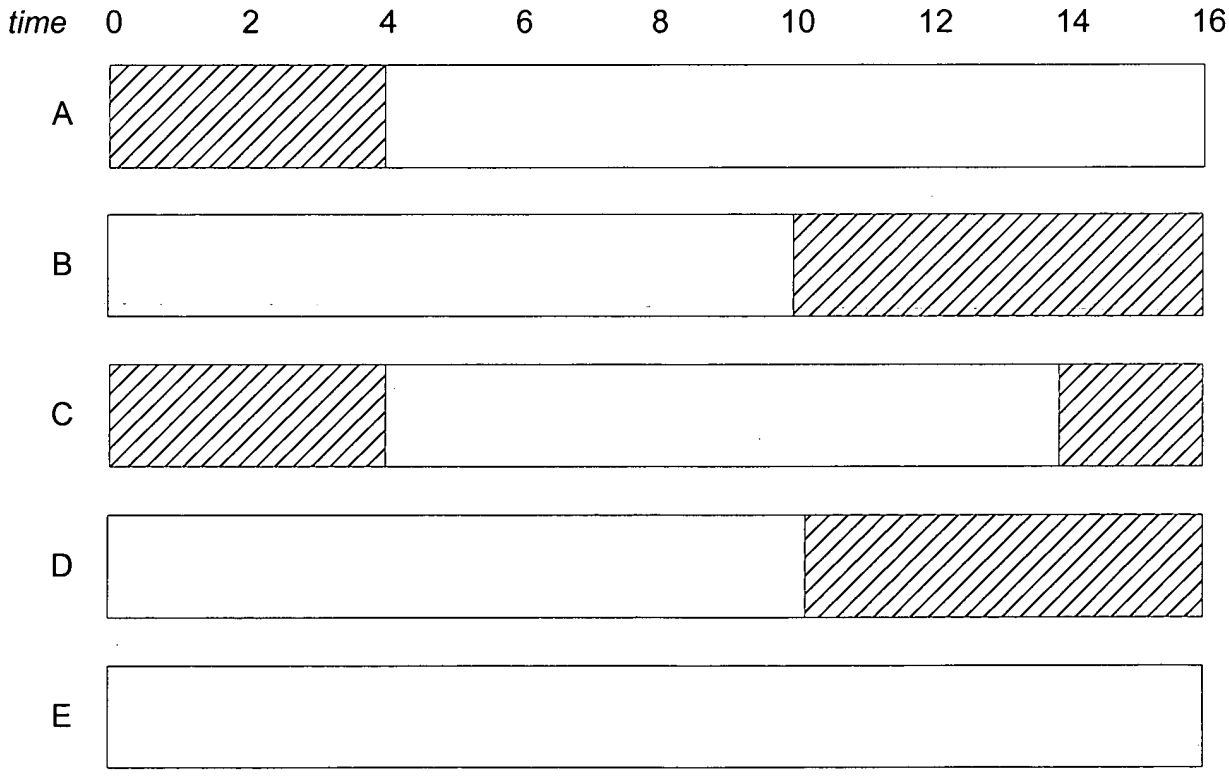

time
0
2
4
6
8
10
12
14
16
A
B
C
D
E

Fig. 26

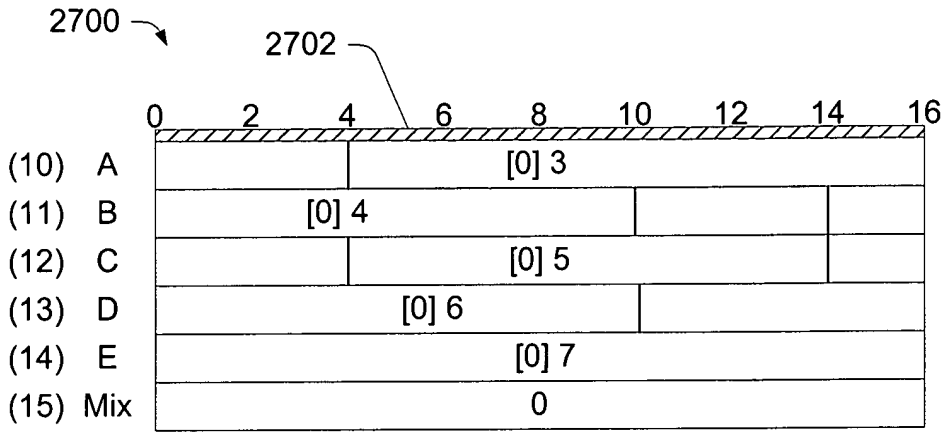

2700
2702
0
2
4
6
8
10
12
14
16
(10) A
(11) B
(12) C
(13) D
(14) E
(15) Mix
[0] 3
[0] 4
[0] 5
[0] 6
[0] 7
0

Fig. 27

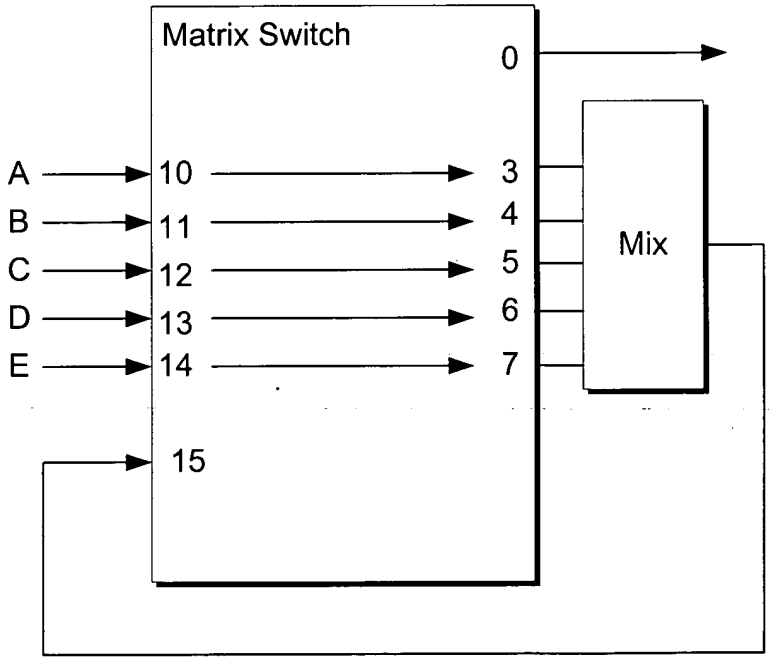

Matrix Switch
A
B
C
D
E
10
11
12
13
14
15
0
3
4
5
6
7
Mix

Fig. 28

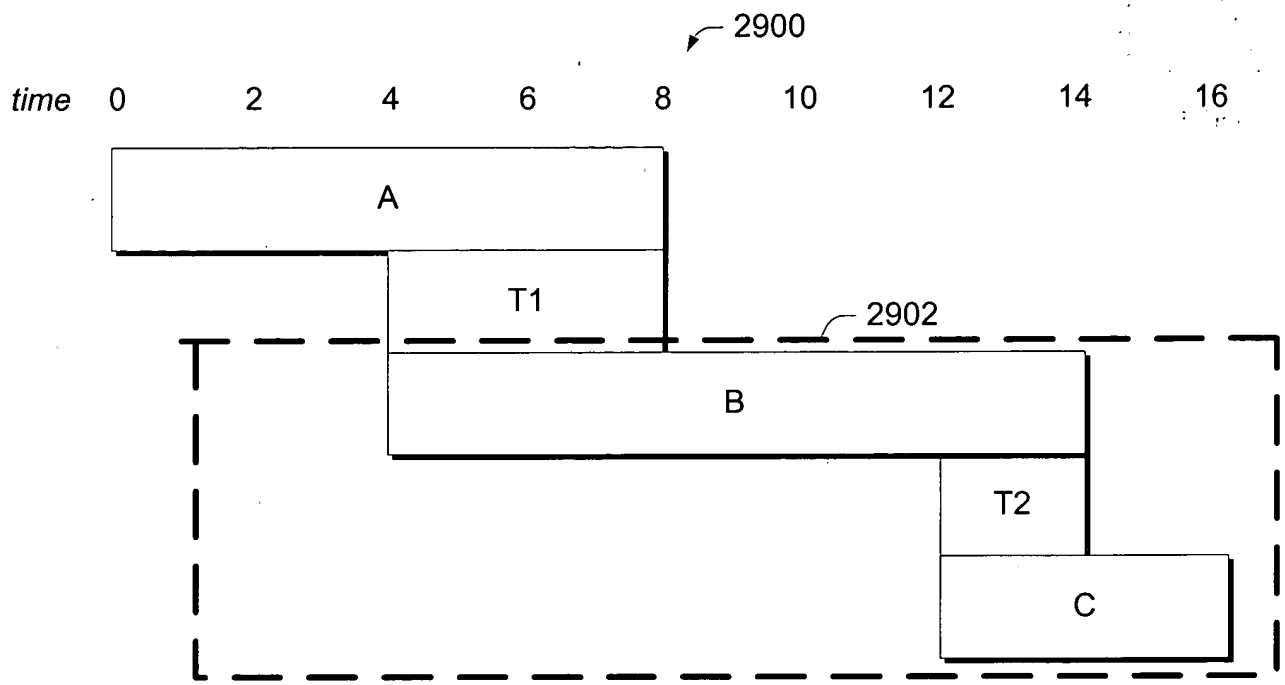

2900
time
0
2
4
6
8
10
12
14
16
A
T1
2902
B
T2
C

Fig. 29

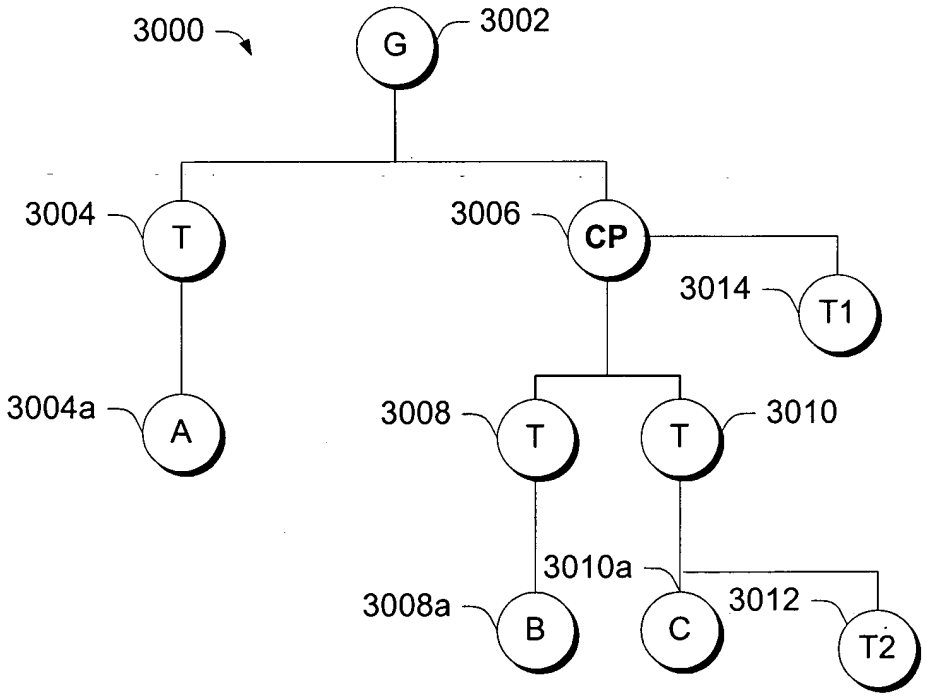

3000
G
3002
3004
T
3006
CP
3014
T1
3004a
A
3008
T
T
3010
3010a
3008a
B
C
3012
T2

Fig. 30

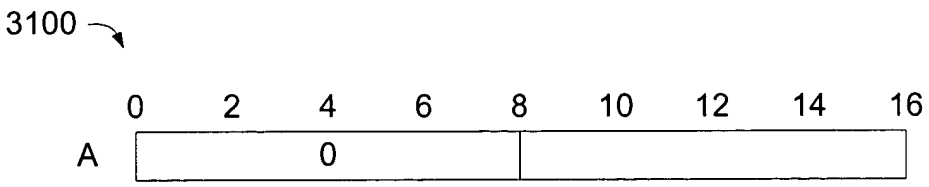


Fig. 31

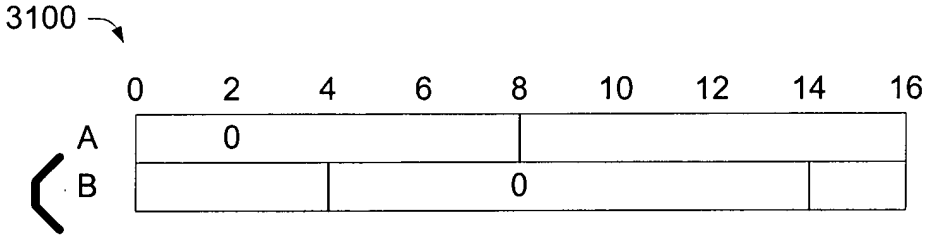


Fig. 32

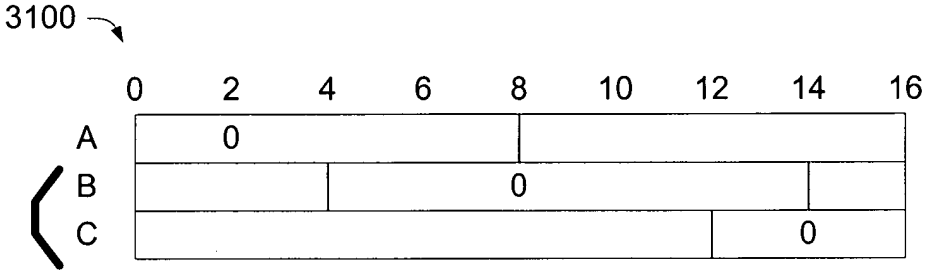


Fig. 33

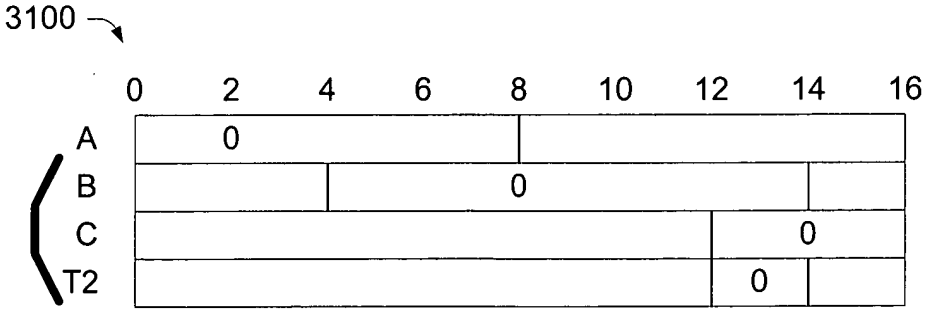


Fig. 34

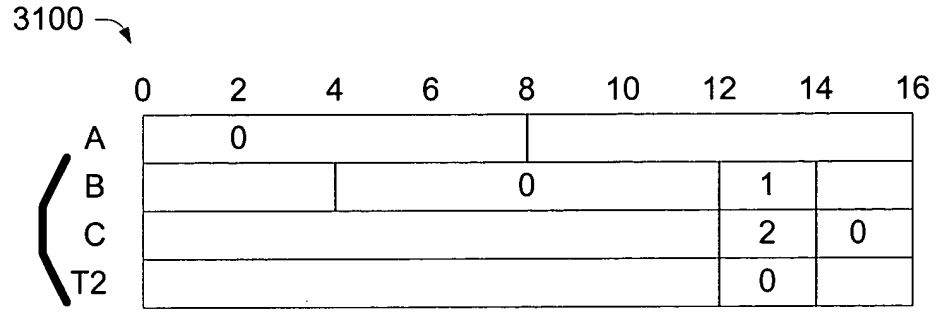


Fig. 35

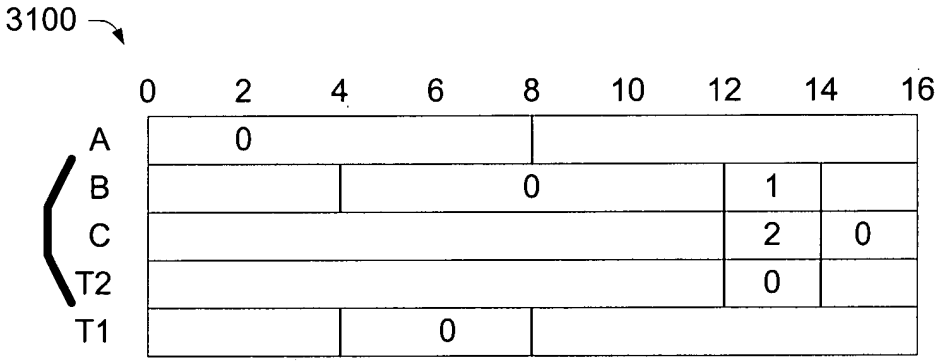


Fig. 36

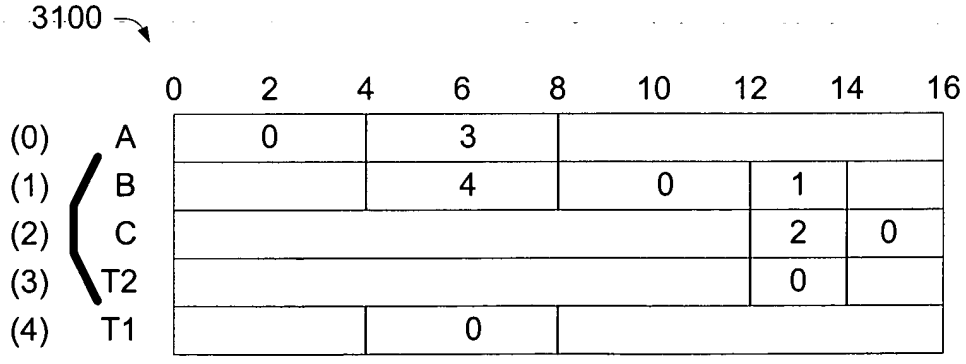


Fig. 37

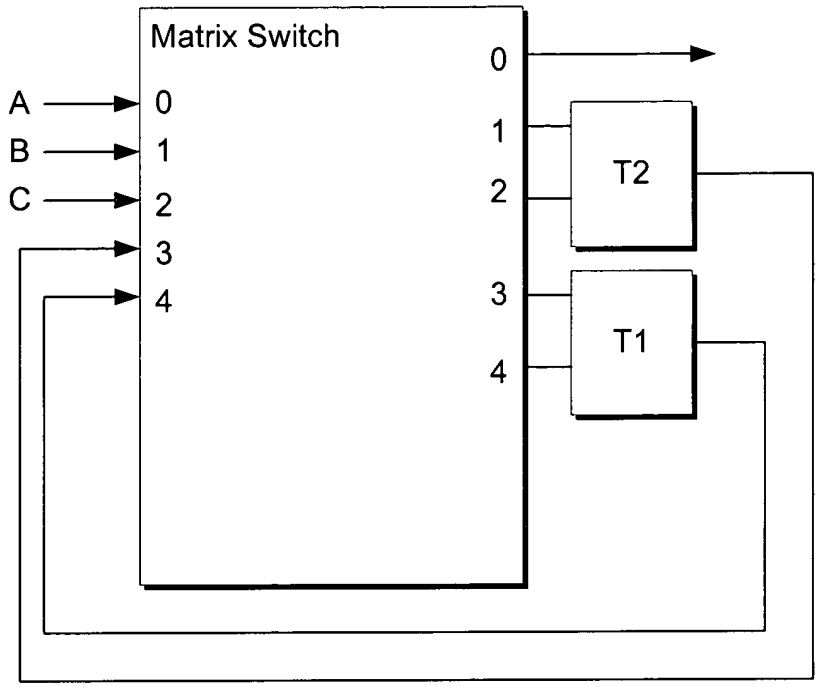


Fig. 38

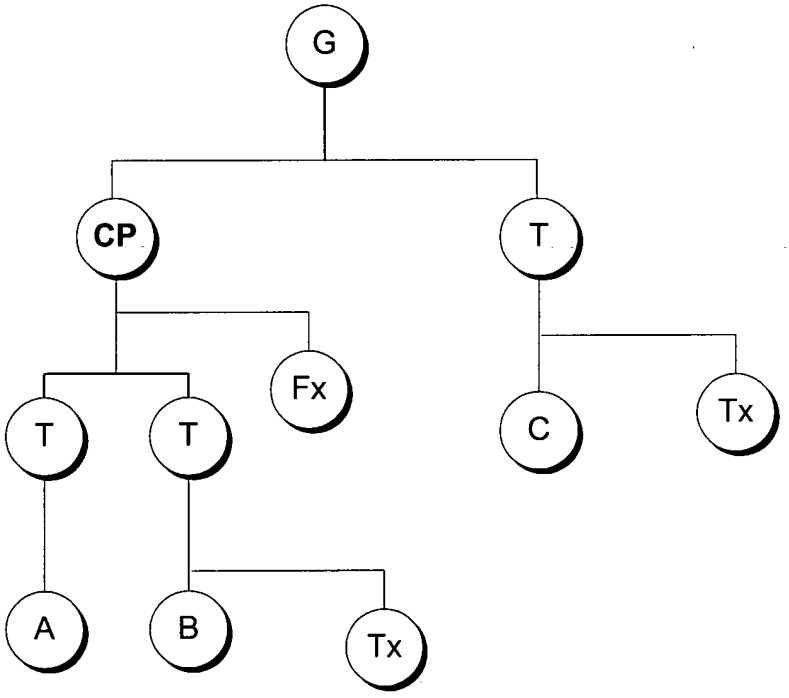


Fig. 38a

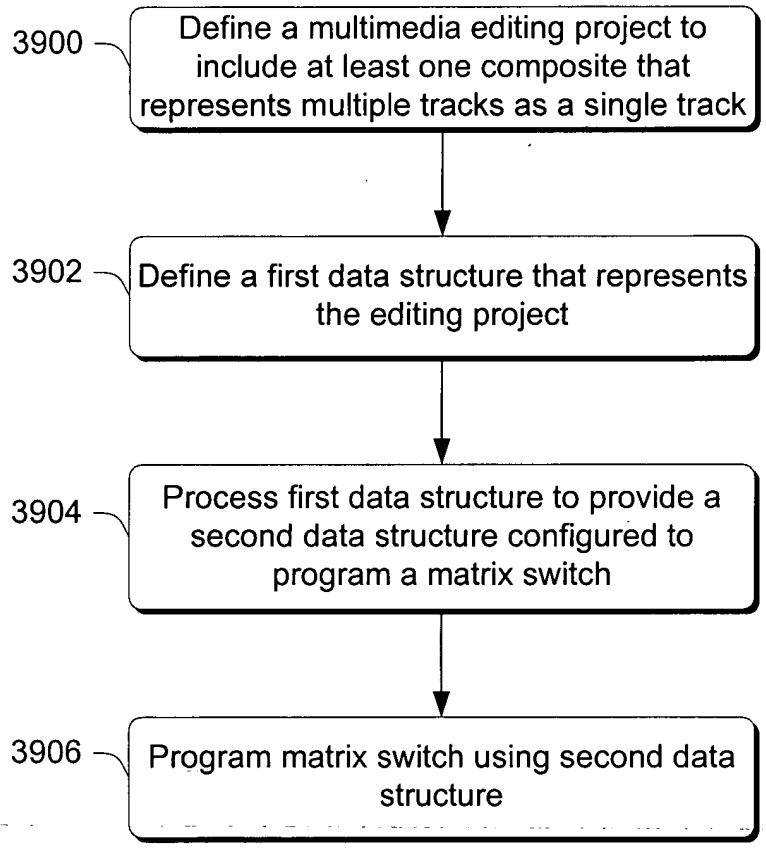

3900
Define a multimedia editing project to include at least one composite that represents multiple tracks as a single track
3902
Define a first data structure that represents the editing project
3904
Process first data structure to provide a second data structure configured to program a matrix switch
3906
Program matrix switch using second data structure

Fig. 39

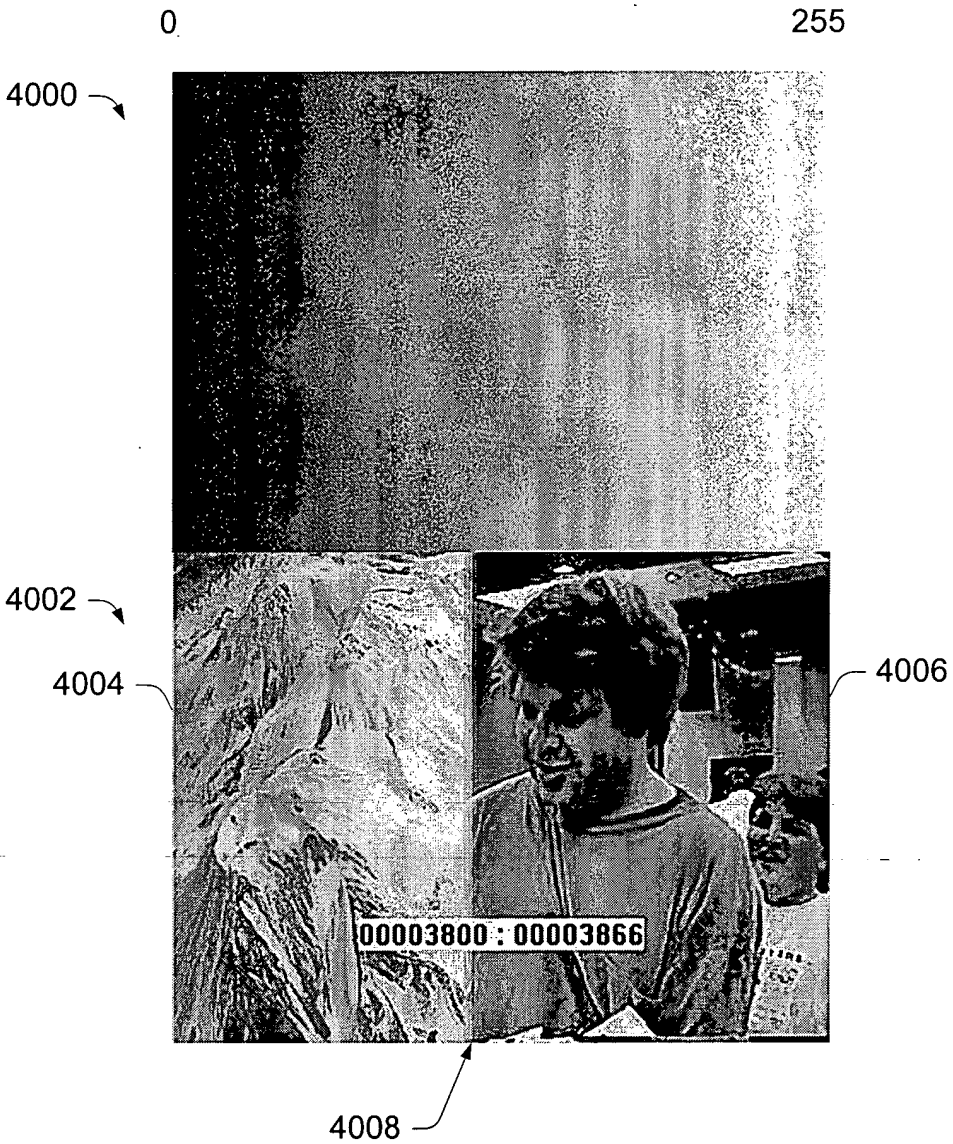


Fig. 40

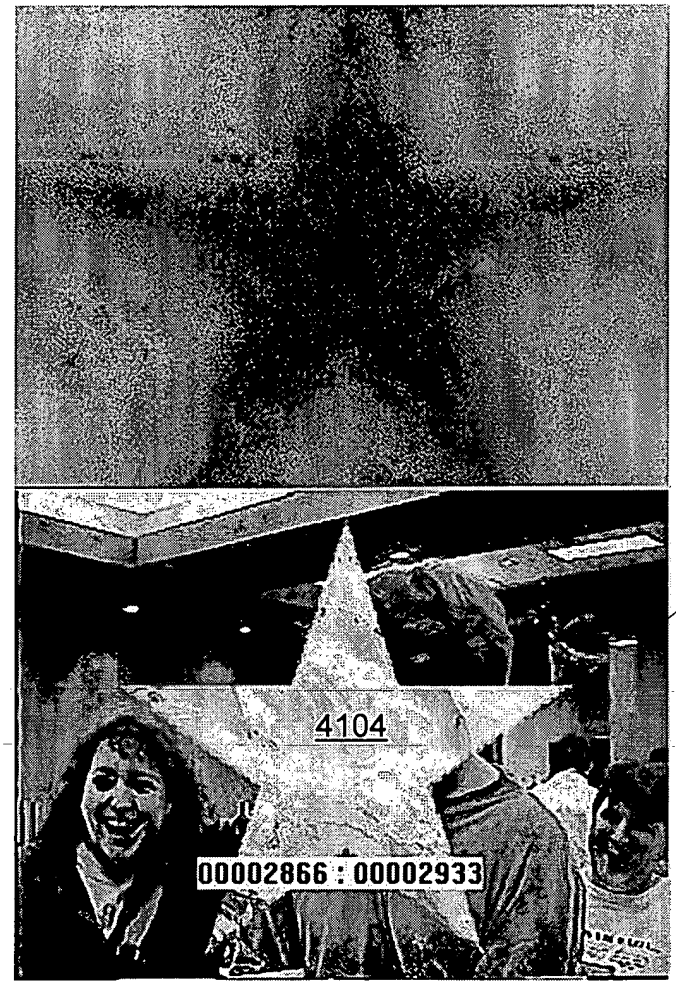


Fig. 41

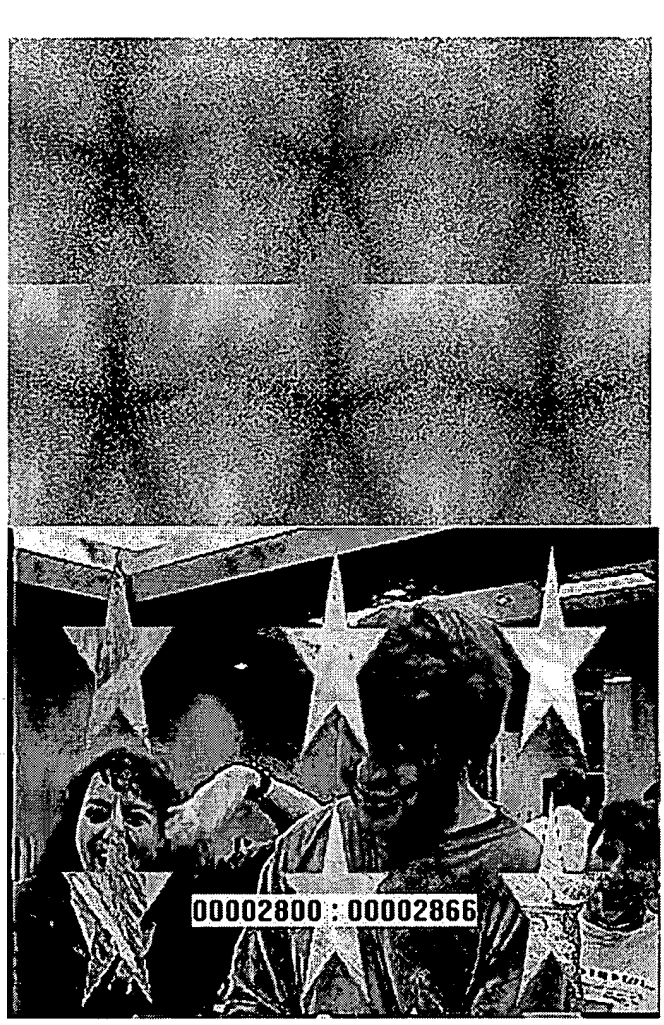


Fig. 42

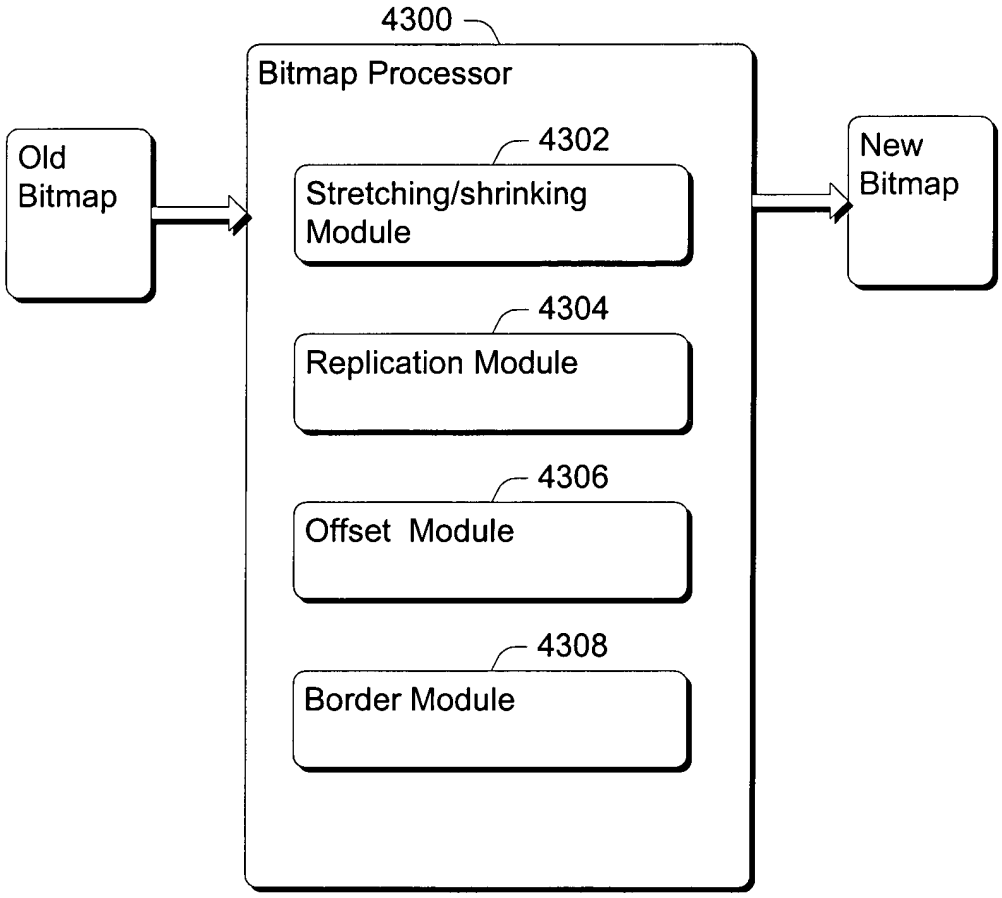

4300
Bitmap Processor
Old Bitmap
4302
Stretching/shrinking Module
4304
Replication Module
4306
Offset Module
4308
Border Module
New Bitmap

Fig. 43

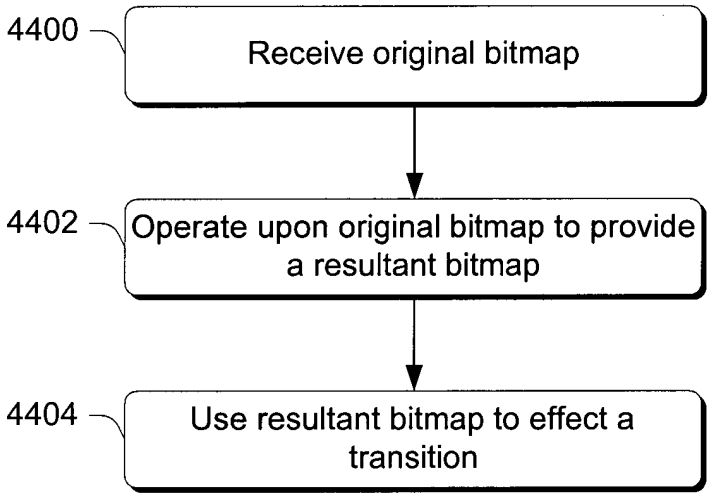

4400
Receive original bitmap
4402
Operate upon original bitmap to provide a resultant bitmap
4404
Use resultant bitmap to effect a transition

Fig. 44

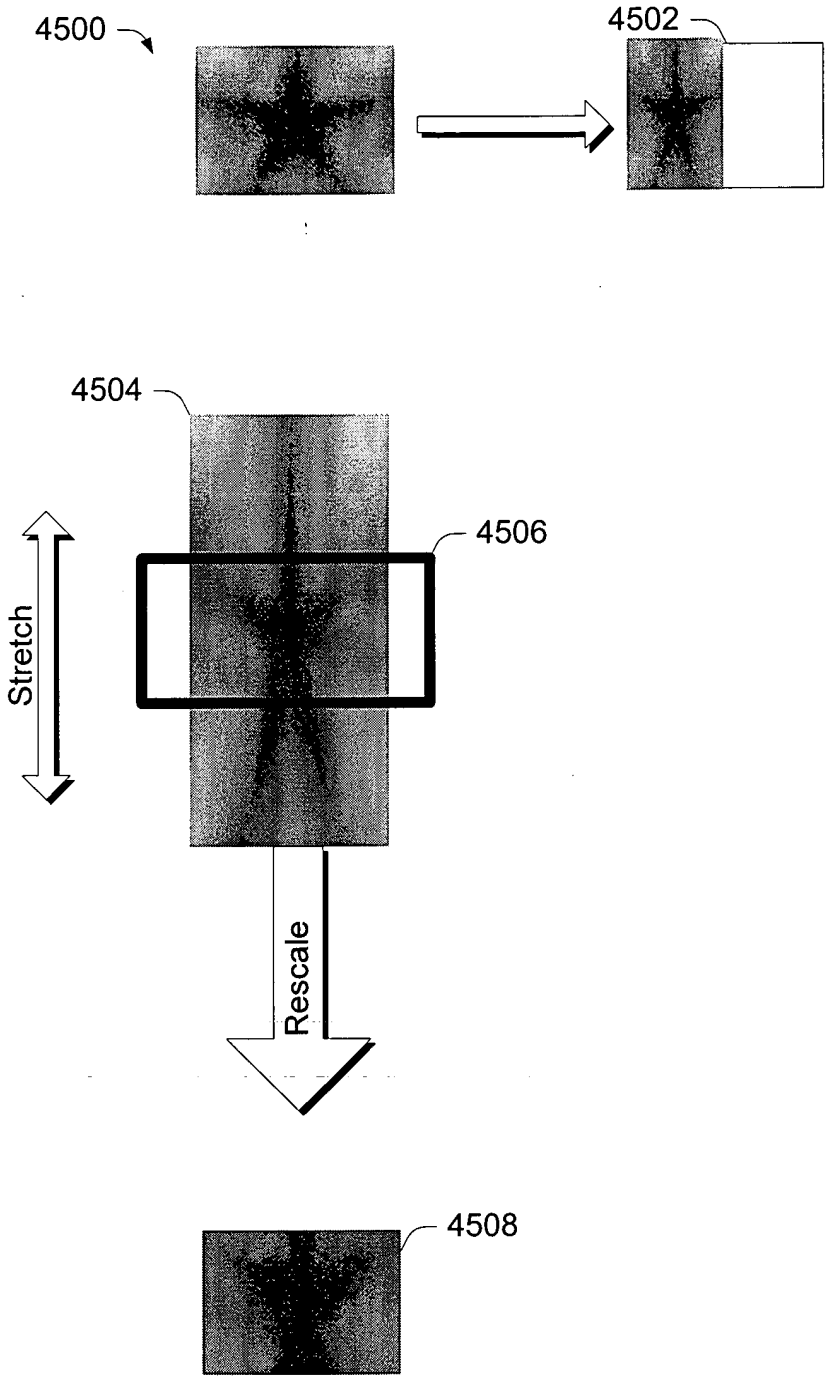


Fig. 45

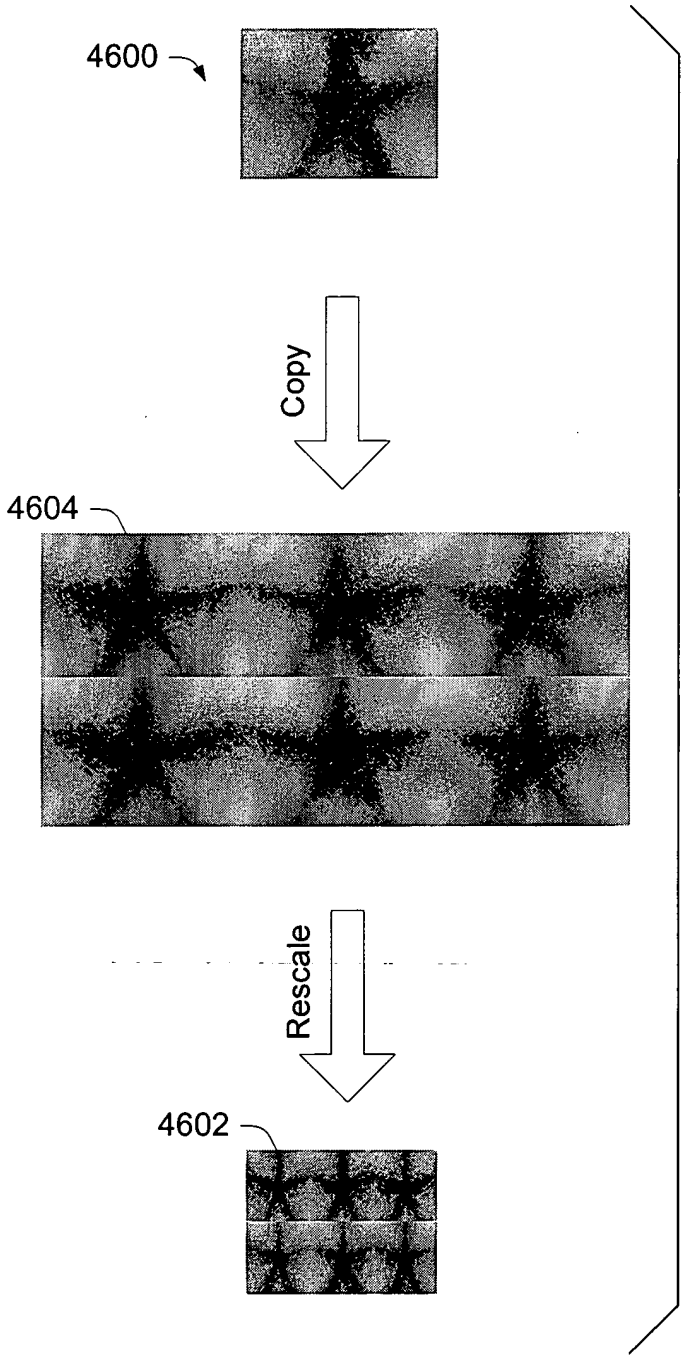


Fig. 46

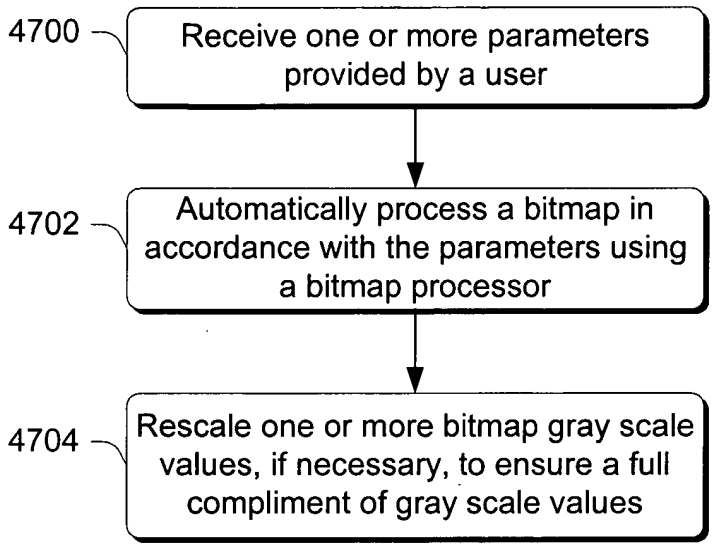

4700
Receive one or more parameters provided by a user
4702
Automatically process a bitmap in accordance with the parameters using a bitmap processor
4704
Rescale one or more bitmap gray scale values, if necessary, to ensure a full compliment of gray scale values

Fig. 47

4800
4802
00002866 : 00002933

Fig. 48

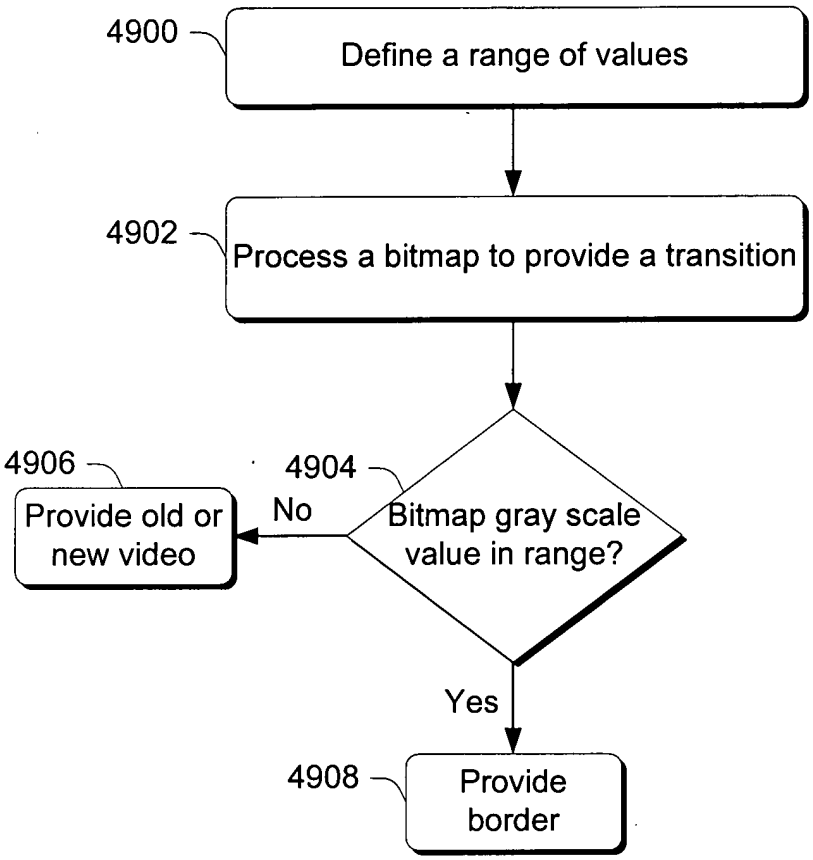

4900
Define a range of values
4902
Process a bitmap to provide a transition
4904
Bitmap gray scale value in range?
No
4906
Provide old or new video
Yes
4908
Provide border

Fig. 49